Arnav Dixit, Aadi Sudan
Professor Ghosh
CSCI 184
June 10, 2025

Predicting Diabetes Using Background User Info

Background:

Diabetes rates in the United States continue to climb. According to the International Diabetes Federation, 11.1% (1 in 9 adults) of the adult population in the US has Diabetes. By 2050, the number is expected to be 1 in 8 adults. In summary, Diabetes is a disease wherein the body can't produce enough insulin to break down sugar or wherein the body doesn't know how to break down sugar with the insulin it does produce. Factors that lead to Diabetes include underlying genetic trends, physical inactivity, and an unhealthy diet. Particularly, trends that are increasing Type 2 Diabetes include urbanization, an aging population, decreasing levels of physical activity, and increasing overweight and obesity prevalence.

Many studies have explored using machine learning to predict diabetes using prior medical information. In 1988, researchers used an early neural network model (ADAP) to create a diabetes classification model (Smith 88). More recently, Kavakiotis et al. (2017) examined ML techniques for diabetes research. The study found that SVM, DecisionTrees, and Logistic Regression are effective for disease classification and identifying risk factors.

The dataset we derived contains annual survey data derived from the CDC (Behavioral Risk Factor Surveillance System), documenting self-reported responses from telephone surveys on chronic health conditions. The responses on the CSV are from the 2015 survey, which contains 253,681 respondents. The dataset has 21 columns: HighBP, HighChol, CholCheck, BMI, Smoker, Stroke, HeartDiseaseorAttack, PhysActivity, Fruits, Veggies, HvyAlcoholConsump, AnyHealthcare, NoDocbcCost, GenHlth, MentHlth, PhysHlth, DiffWalk, Sex, Age, Education, and Income.

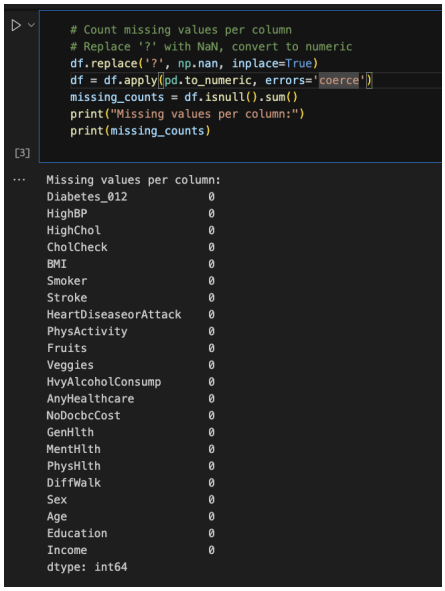
Design:

The goal of this study is to successfully classify instances where an individual may have diabetes based on other data about their life and health. All features (except for the Diabetes classification column) will be used for the input. We will use multiple models to test classification: RandomForest, Neural Networks (softmax activation function for output layer), DecisionTree, KNN, and Logistic Regression as those are fit for classification. From there, we will test for the precision, recall, and f1 score of each of the models, along with test accuracy. We will also use CrossValidation, SVM (LinearSVC), and Naive Bayes (GaussianNB). The model with the highest accuracy will be used for predicting new instances. We will use the model for a

full-stack application, inputting the prediction along with relevant medical context into an Ollama LLM-equipped RAG to output a medical report regarding the classification (preliminary diagnosis).

Implementation:

We loaded the Diabetes dataset CSV with pandas and proceeded to then display it to confirm successful loading. We then proceeded to find how many missing values there were per column, wherein we found no column with null values.

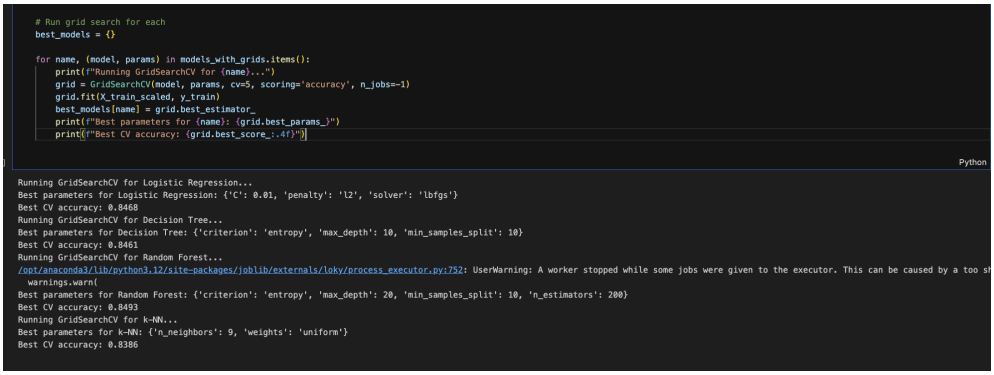


```
# Count missing values per column
# Replace '?' with NaN, convert to numeric
df.replace('?', np.nan, inplace=True)
df = df.apply(pd.to_numeric, errors='coerce')
missing_counts = df.isnull().sum()
print("Missing values per column:")
print(missing_counts)
```

```
[3]
...  Missing values per column:
Diabetes_012            0
HighBP                  0
HighChol                0
CholCheck               0
BMI                     0
Smoker                  0
Stroke                  0
HeartDiseaseorAttack    0
PhysActivity            0
Fruits                  0
Veggies                 0
HvyAlcoholConsump       0
AnyHealthcare           0
NoDocbcCost             0
GenHlth                 0
MentHlth                0
PhysHlth                0
DiffWalk                0
Sex                     0
Age                     0
Education               0
Income                  0
dtype: int64
```

The feature we were trying to set as the designated output is the Diabetes_12 column, which is a 0 to 1 value which indicates whether the profile in the designated row has diabetes or not. All the other features are set as input features.

We chose to test prediction accuracy on several classification models due to their ability to accurate classify binary data - including Logistic Regression, Decision Trees, Random Forest, KNNs, SVMs, and Naive Bayes. From there, we ran grid search for all models except for SVM and Naive Bayes to get the best hyperparameters for the models.



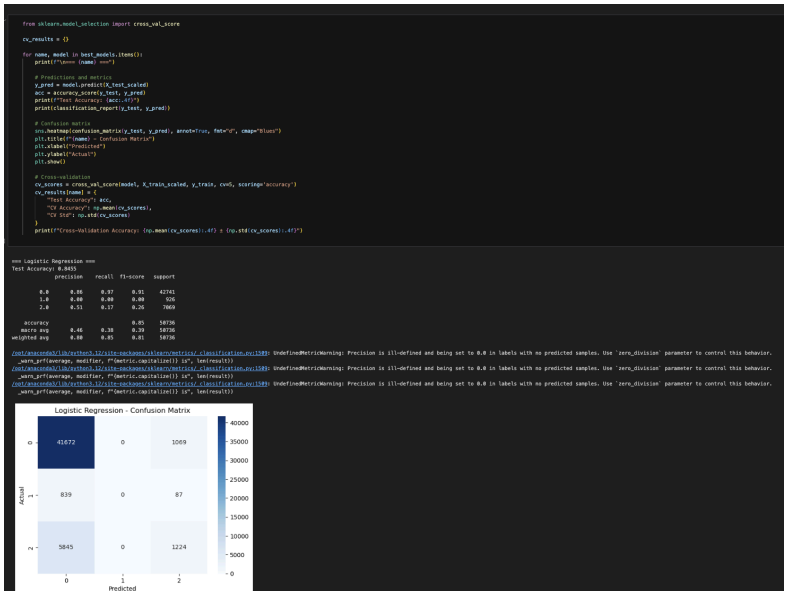
```
# Run grid search for each
best_models = {}

for name, (model, params) in models_with_grids.items():
    print(f"Running GridSearchCV for {name}...")
    grid = GridSearchCV(model, params, cv=5, scoring='accuracy', n_jobs=-1)
    grid.fit(X_train_scaled, y_train)
    best_models[name] = grid.best_estimator_
    print(f"Best parameters for {name}: {grid.best_params_}")
    print(f"Best CV accuracy: {grid.best_score_:.4f}")
```

Python

```
Running GridSearchCV for Logistic Regression...
Best parameters for Logistic Regression: {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
Best CV accuracy: 0.8468
Running GridSearchCV for Decision Tree...
Best parameters for Decision Tree: {'criterion': 'entropy', 'max_depth': 10, 'min_samples_split': 10}
Best CV accuracy: 0.8461
Running GridSearchCV for Random Forest...
/opt/anaconda3/lib/python3.12/site-packages/joblib/externals/loky/process_executor.py:752: UserWarning: A worker stopped while some jobs were given to the executor. This can be caused by a too sh
  warnings.warn(
Best parameters for Random Forest: {'criterion': 'entropy', 'max_depth': 20, 'min_samples_split': 10, 'n_estimators': 200}
Best CV accuracy: 0.8493
Running GridSearchCV for k-NN...
Best parameters for k-NN: {'n_neighbors': 9, 'weights': 'uniform'}
Best CV accuracy: 0.8386
```

After hyperparameter tuning, we proceeded to run the models on the dataset to identify the model with the highest test accuracy, while recording precision (measures accuracy of positive predictions) recall (measuring accuracy of negative predictions), and F1 score (combines precision and recall to measure model performance). We used seaborne to create a confusion matrix, which outlines true positives, false positives, false negatives, and true negatives. We also used cross-validation on each of the models to unseen data.

Example of output from Logistic Regression (confusion matrix with seaborne)

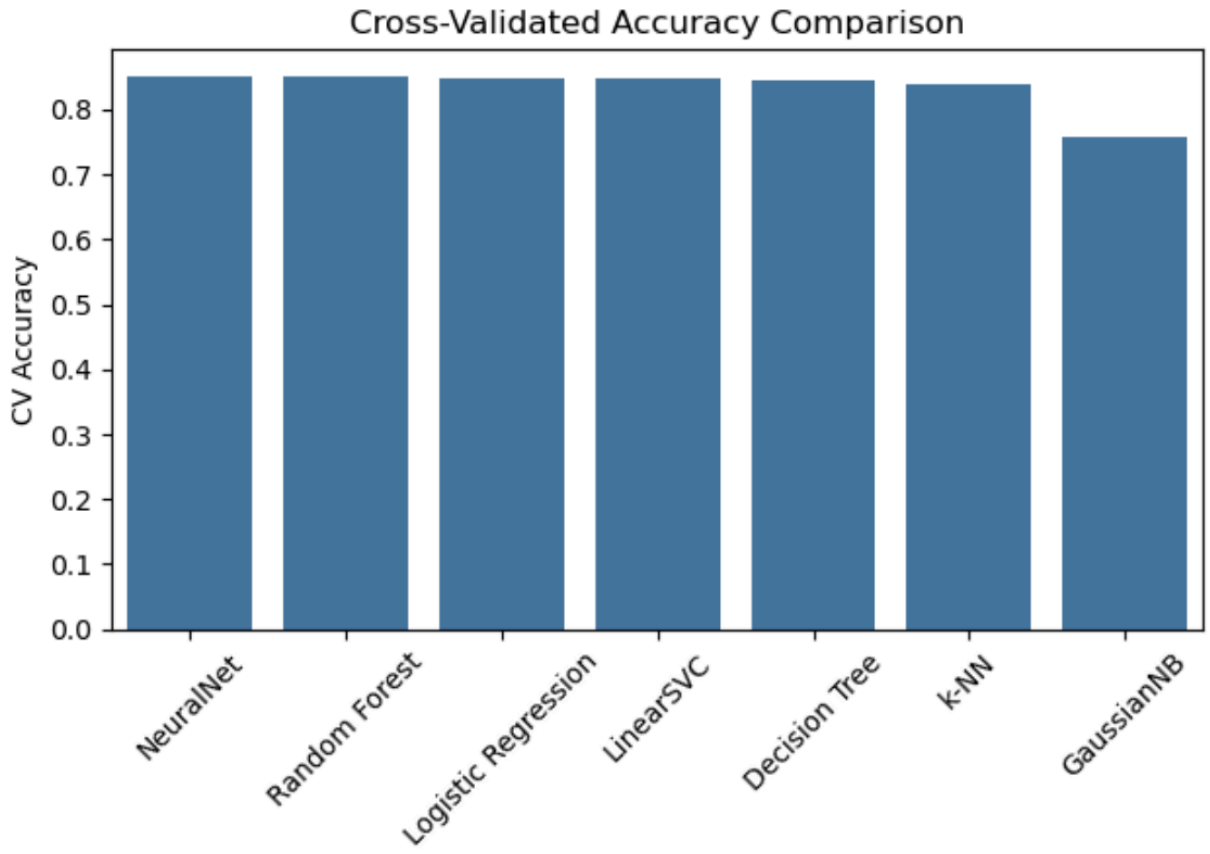


We also used Neural Networks for binary classification, one-hot-encoding the output (into one-hot vectors). This is required since our output layer uses softmax and the loss function is cross entropy. The neural network has an input layer of 1 neuron, 2 hidden layers of 64 neurons, and an output layer of 3 neurons to output class probability. We used a confusion matrix and K-folds cross validation to evaluate the Neural Network model we developed.

Results:

After training and evaluating all selected models, we compiled both test accuracy and cross-validation scores into a comparative summary. The dataset allowed for robust model training and meaningful performance evaluation due to a dimensionality of over 200,000 samples. The following results reflect each model's performance across unseen test data, as well as an internal estimate of its generalizability using k-fold cross-validation with 5 folds.

```
Model Comparison Summary:
                     Test Accuracy  CV Accuracy    CV Std
NeuralNet                 0.846480     0.849377  0.001048
Random Forest             0.849850     0.849254  0.000709
Logistic Regression       0.845475     0.846775  0.000875
LinearSVC                 0.845869     0.846362  0.000977
Decision Tree             0.845652     0.846110  0.000517
k-NN                      0.839857     0.838596  0.000887
GaussianNB                0.757884     0.757002  0.002319
```



Interpretation:

The results suggest that multiple ML models are capable of accurately predicting diabetes status using only self-reported health and demographic data. The small spread in test accuracy among Random Forest, Neural Network, Logistic Regression, and LinearSVC indicates that the problem space is well-structured and relatively linearly separable.

The success of the Random Forest model aligns with its ability to handle high-dimensional tabular data and automatically model interactions between features - our data has a hefty dimensionality of over 250,000 samples and 22 features. Its extremely low cross-validation standard deviation shows strong generalization across folds, confirming robustness. The Neural Network performed slightly better in cross-validation than in test accuracy, indicating a potential for improved generalization with more training or additional

tuning. Adding more hidden layers or neurons could possibly help improve the model's capabilities, though testing the optimal build for this network would be a costly and time-consuming endeavor

Logistic Regression and LinearSVC showed strong performance with fast training and easy interpretability, making them excellent options for real-world deployment where transparency is important. Their high accuracy confirms that much of the decision boundary is linear.

The Decision Tree performed nearly as well as its ensemble version but had slightly more variance, as expected for single estimators. K-NN had modest performance and would not scale well in a real-time application due to slow inference and poor generalization. Naive Bayes underperformed, likely due to its assumption of feature independence, which does not hold in health datasets where features such as activity, weight, and diet are often interrelated and showcase strong correlation with the target only when put together.

It's worth noting the remarkably low standard deviations for cross-validation accuracy across all models (except GaussianNB), suggesting consistent behavior across the five folds. This low variance across partitions implies that the dataset is not only large, but also well-stratified and diverse, allowing models to learn generalizable patterns without overfitting to specific subsets of the data.

Conclusion:

This study demonstrates that machine learning models can predict diabetes status with high accuracy using only background user data from large-scale public health surveys. With models like Random Forest and Neural Networks achieving nearly 85% accuracy, the results are on par with — or even better than — some clinical screening tools, and they require no invasive testing.

In practical terms, this means that health organizations, public clinics, or even individuals could use simple web forms to gather survey-style data and get a reasonably accurate prediction of diabetes risk. Because our best models use only self-reported, non-clinical features, the approach is highly scalable and accessible to populations that may lack consistent medical access.

We plan to deploy our best-performing model, the Random Forest, onto a full-stack web application that uses RAG (retrieval-augmented generation) techniques to return contextualized medical reports powered by a local LLM. The app will take survey inputs, classify the user's diabetes risk, and supplement the result with medically relevant educational content. Future improvements could include model explainability using SHAP values, deeper neural network

experimentation, or feature augmentation with wearable sensor data (e.g., step count, sleep tracking).

Sources:
https://idf.org/about-diabetes/diabetes-facts-figures/
https://github.com/lorilist/pima-indians-diabetes-prediction
https://pubmed.ncbi.nlm.nih.gov/28138367/